

# 2020년도 2학기 자료구조

## HW1 결과보고서

20190345 김동현

### 1. 과제 요약

문제1은 iterative and recursive 두 가지 방법으로 Binomial Coefficient를 계산하는 프로그램을 구현하는 것이다. 문제2는  $N \times M$ 크기의 직사각형을 한번의 길이가 2의 제곱인 정사각형을 재귀적으로 최소한으로 사용해서 채우는 것이다.

### 2. 과제 환경

CSPRO 환경에서 실습을 진행하였다.

CSPRO 서버 : cspro.sogang.ac.kr

### 3. 과제 구현 내용

문제1은 다음과 같은 크게 입력부분, 재귀적 이항계수 계산, 비재귀적 이항계수 계산으로 구성된다. 변수 선언과 입력 부분은 다음과 같다.

```
int n,m,result;

scanf("%d %d", &n, &m);
if(n/100000000000>=1||m/100000000000>=1){
    printf("input error");
    return 0;
}
```

사용자에게 입력받을 정수형 변수 N,M을 선언한 뒤, scanf를 이용해 입력 받는다. 이때 제약 조건인 입력은 최대 10글자를 넘지 않는다는 조건을 실행하기 위해 조건문을 사용한다. 10의 11제곱으로 나눈 값이 0보다 크다면 이는 10자리보다 크다는 말이므로 error를 출력한 뒤, 종료한다.

재귀적 계산은 다음과 같은 함수로 구현한다.

```
int rBC(int n, int m){
    if(n==m||m==0){
        return 1;
    }
    else {
        return rBC(n-1,m)+rBC(n-1,m-1);
    }
}
```

우선 이항계수로 들어온 두 수를 살펴보면 다음과 같다. N개의 서로 다른 것 중에서 M개를 선택하는 경우의 수이다. 따라서 N과 M이 같으면 모두 다 선택하는 1가지 경우의 수이므로 1을 리턴한다. 또한  $[N,M]=[N-1,M]+[N-1,M-1]$ 로 정의할 수 있다. 이때 N은 항상 M보다 커야하므로 재귀적으로 반복을 진행하였을 때, N보다 M이 먼저 0에 도달한다. 따라서 N개의 중 0개를 고르는 경우는 모두 고르지 않는 1가지 경우이므로 1을 리턴한다. 반면 위의 두 경우가 아닐 경우에는 다시 rBC함수를 호출한다. 이때의 인자는 각각 N-1,M과 N-1,M-1이다. N-1,M의 경우는 N개의 경우의 수 하나가 반드시 포함되지 않았다는 것을 전제로 한다. 따라서 나머지 N-1개 중 M개를 선택하는 경우이다. 이와 반대되는 것이 N개 중 1개를 반드시 포함하는 것을 전제로 한다. 따라서 N-1개 중 M-1개를 선택하는 것이다. 이렇게 진행할 경우 인자로 주어진 N과 M값이 점점 종료 조건인 0으로 다가가기 때문에 재귀적으로 결과를 호출할 수 있다.

비재귀적 표현은 다음과 같이 구성한다.

```
int BC(int n, int m){
    int a=1;
    int i;
    for(i=n;i>n-m;i--){
        a*=i;
    }
    for(i=m;i>0;i--){
        a/=i;
    }

    return a;
}
```

이항계수를 비재귀적으로 살펴보면 다음과 같다.  $\left[ \begin{matrix} N \\ M \end{matrix} \right] = \frac{N!}{M!(N-M)!}$ 로 표현이 가능하다. 따라서 N!값에서 M!과 (N-M)!의 값을 나누면 값을 구할 수 있다. 하지만 연산과정을 줄이기 위해 약분을 고려할 수 있다. 이항계수의 정의에 따라 N은 M보다 크거나 같다. 따라서 N!을 (M-N)!로 나누더라도 그 값은 항상 정수가 되고, 그 값은 N에서 N-M+1까지의 연속된 정수의 곱과 같다. 이를 통해  $\frac{N!}{(N-M)!}$ 을 계산한다. 이후 M!값을 다시 나누어주면 결과값이 나온다.

문제2는 입력 부분과 재귀적 계산 부분으로 나뉜다. 변수를 선언하고 입력 받는 부분은 다음과 같다.

```
int N,M;
int count;
scanf("%d %d",&N,&M);
if(N/10000000000>1||M/10000000000>1){
    printf("input error");
    return 1;
}
```

가로길이를 N, 세로길이를 M으로 입력을 받는다. 정수형 변수 count는 필요한 타일의 개수이다. 만약 입력받은 값이 10자리가 넘어간다면 입력을 종료한다.

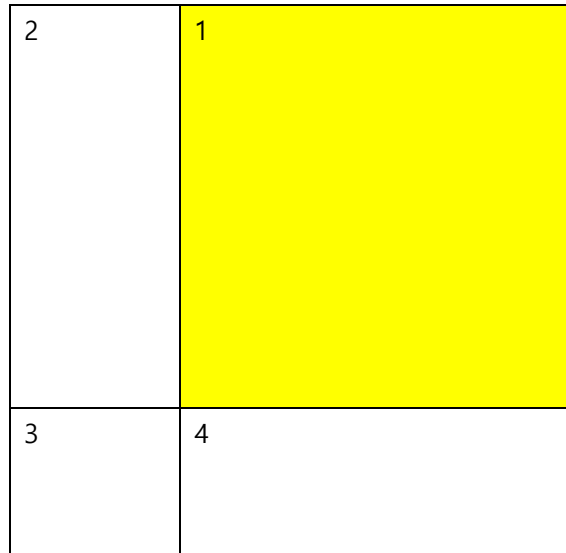
재귀적인 표현은 다음과 같다.

```
int rtile(int N, int M){
    int n,m,i,count;
    for(i=1;i<=N&&i<=M;i*=2){
        ;
    }
    if(i>1){
        i/=2;
    }
    count=(N/i)*(M/i);
    n=(N/i)*i;
    m=(M/i)*i;

    if(N-n<=1){
        if(M-m<=1){
            count+=(N-n)*(M-m);
            count+=(N-n)*m;
            count+=n*(M-m);
            return count;
        }
        else{
            count+=(N-n)*M;
            return count + rtile(n,M-m);
        }
    }
    else if(M-m<=1){
        count+=N*(M-m);
        return count+rtile(N-n,m);
    }
    else{
        return count+rtile(N-n,M)+rtile(n,M-m);
    }
}
```

재귀함수 rtile은 인자로 메인함수에서 가로의 길이와 세로의 길이를 각각 N,M으로 받는다. 이후 정수형 변수 i는 N과 M을 넘지않는 2의 제곱수의 최대값을 의미한다. 만약 N이 20이고 M이 25 라면 i는 16이 되어야한다. 하지만 반복문 안에서는 i의 값이 32가 된다. 따라서 반복문이 종료되

고  $i$ 값을 2로 나눠준다. 그리고  $N \times M$ 의 사각형을 두 부분으로 나눌 수 있다.  $N, M$ 을 넘지 않는 2의 제곱수의 최대값을 한 변으로 하는 정사각형이 들어갈 수 있는 부분과, 들어갈 수 없는 부분으로 나뉜다. 또한 들어갈 수 없는 부분 또한 들어갈 수 있는 부분의 꼭짓점을 기준으로 3등분으로 나눌 수 있다. 다음과 같은 직사각형이 만들어진다. 노란색 부분이 정사각형이 들어갈 수 있는 부분이고, 하얀색 부분이 그렇지 않은 부분이다. 각각의 사각형에 1부터 4까지의 번호를 부여한다.



2번 사각형은 한 변의 길이가  $i$ 인 정사각형으로 이루어져있다. 이때의 길이는 가로, 세로 각각  $N$  또는  $M$ 을 넘지 않는  $i$ 의 배수값이다. 따라서 사각형의 개수는  $(N/i) \times (M/i)$ 가 된다.  $(N/i)$ 는 가로에 들어간 정사각의 개수,  $(M/i)$ 는 세로에 들어간 정사각형의 개수이다. 둘을 곱하면 1번 부분에 들어갈 정사각형의 개수가 된다. 이후 2,3,4번을 하나의 직사각형을 보며, `rtile` 함수를 재귀적으로 불러오면 된다. 변수  $n, m$ 은 기존 가로와 세로 길이  $N, M$ 에서 1번 부분의 가로와 세로 길이를 뺀 값이다. 이후 멈추는 조건은 하나의 인자가 1일 경우이다. 이때는 한 변의 길이를 계산할 필요없이 반드시 1이 나오는데, 이럴 경우 인자 둘의 곱이  $1 \times 1$ 인 정사각형의 개수가 되므로, 함수를 불러올 필요없이 바로 `count` 변수에 더한다. 결국 모든 위치에 정사각형이 들어오거나, 한 변의 길이가 1이 된다면, 재귀함수를 종료하고, 정사각형의 개수 `count`를 리턴한다.